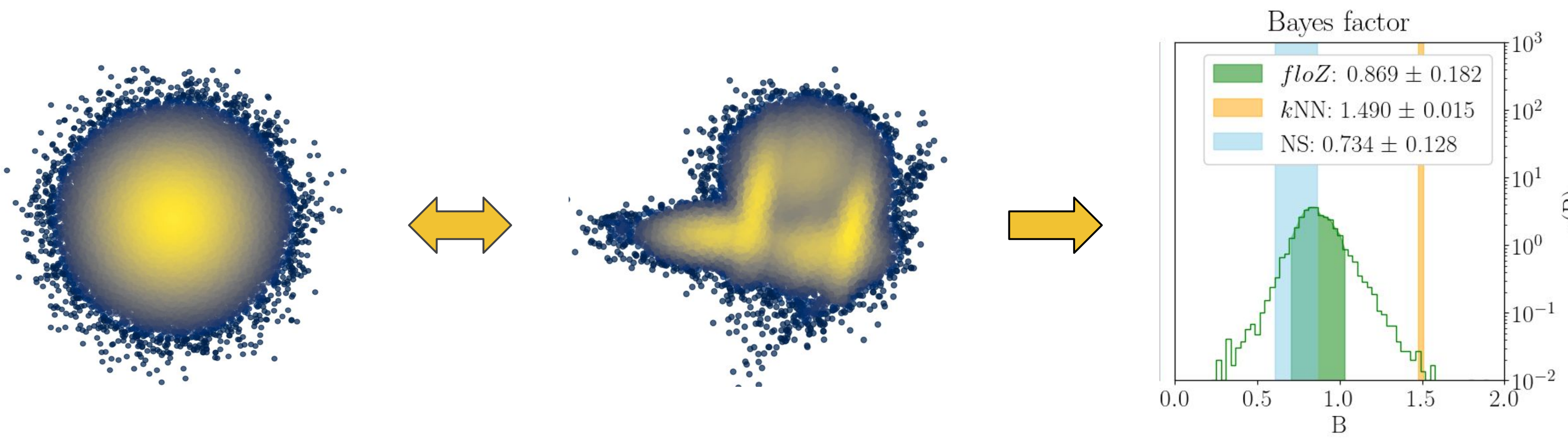
floZ

Bayes factor from normalizing flows

Rahul **Srinivasan**, Marco Crisostomi, Roberto Trotta, Enrico Barausse, and Matteo Breschi



Istituto Nazionale di Fisica Nucleare

1st TEONGRAV international workshop on theory of gravitational waves

arXiv:2404.12294v2; submitted to PRD
https://github.com/Rahul-Srinivasan/floZ

The Bayes Theorem

$$p(A|\mathrm{B}) = \frac{p(\mathrm{B}|\mathrm{A}) \cdot p(A)}{p(\mathrm{B})}$$

The Bayes Theorem

$$p(A|\mathrm{B}, \mathrm{C}) = \frac{p(\mathrm{B}|\mathrm{A}, \mathrm{C}) \,.\, p(A|\mathrm{C})}{p(\mathrm{B}|\mathrm{C})}$$

The Bayes Theorem

$$p(\theta|\text{data}, M) = \frac{p(\text{data}|\theta, M)\ .\ p(\theta|M)}{p(\text{data}|M)}$$

The Bayes Theorem

$$\textit{Posterior} = \frac{\textit{Likelihood} \;.\; \textit{Prior}}{\textit{Evidence}}$$

Competing models

$$p(\theta|\text{data}, M_1) = \frac{p(\text{data}|\theta, M_1) \cdot p(\theta|M_1)}{p(\text{data}|M_1)}$$

Does the data favour M_1 or M_2?
And by how much?

$$p(\theta|\text{data}, M_2) = \frac{p(\text{data}|\theta, M_2) \cdot p(\theta|M_2)}{p(\text{data}|M_2)}$$

Competing models

$$p(\theta|\text{data}, M_1) = \frac{p(\text{data}|\theta, M_1) \;\cdot\; p(\theta|M_1)}{p(\text{data}|M_1)}$$

$$p(\theta|\text{data}, M_2) = \frac{p(\text{data}|\theta, M_2) \;\cdot\; p(\theta|M_2)}{p(\text{data}|M_2)}$$

The Bayes Factor

By what factor does the data favour M_1 over M_2?

$$B = \frac{p(\text{data}|M_1)}{p(\text{data}|\mathrm{M}_2)}$$

$$= \frac{Evidence_1}{Evidence_2}$$

The Evidence

$$p(\theta|\text{data},M) = \frac{p(\text{data}|\theta,M)\,.\,p(\theta|M)}{p(\text{data}|M)}$$

Probability density
i.e., normalized.

The Evidence

$$p(\theta|\text{data},M) = \frac{p(\text{data}|\theta,M)\ .\ p(\theta|M)}{p(\text{data}|M)}$$

$$p(\text{data}|M) = \int p(\text{data}|\theta,M)\ .\ p(\theta|M)\ d\theta$$

$$Evidence = \int Likelihood\ .\ Prior\ d\theta$$

Computing this integral can be quite non-trivial, and often, intractable.

State of the art and their ***shortcomings***

State of the art and their ***shortcomings***

Nested sampling[1]:

Evidence estimated by iteratively computing the likelihood.

- ***Computationally intensive*** likelihood *re*calculation.
- ***Slow***, CPU calculations, not parallelizable with GPUs.
- ***Scalability*** issues for high dimensions
  - Ex: 150 dimensions are computationally prohibitive

1. John Skilling "Nested Sampling," 10.1063/1.1835238.

State of the art and their ***shortcomings***

Nested sampling[1]:

Evidence estimated by iteratively computing the likelihood.

- ***Computationally intensive*** likelihood *re*calculation.
- ***Slow***, CPU calculations, not parallelizable with GPUs.
- ***Scalability*** issues for high dimensions

Other techniques:

1. k-nearest neighbours[2], Laplace approx. - ***Less expressive***: fails for large non-gaussianity.
2. Normalizing flow-based nested[3]/Gaussianized bridge[4] sampling - ***Requires likelihood re-calculation***

1. John Skilling "Nested Sampling," 10.1063/1.1835238.
2. A. Heavens, et al 2017 arXiv:1704.03472 [stat.CO]
3. Nested sampling with normalizing flows for gravitational-wave inference, 10.1103/PhysRevD.103.103006
4. Jia, He; Seljak, Uroš, 2019 10.48550/arXiv.1912.06073

State of the art and their ***shortcomings***

Nested sampling[1]:

Evidence estimated by iteratively computing the likelihood.

Likelihood evaluation can be expensive.
These are pre-computed for MCMC samples in parameter estimation pipelines.
Why not use it?

Useful to have a <u>fast</u>, <u>scalable</u>, and <u>expressive</u> method that does not require extra likelihood evaluations.

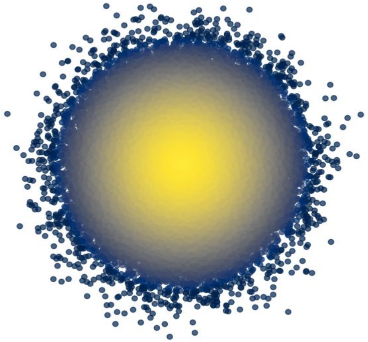
1. k-nearest neighbours[2], Laplace approx. - ***Less expressive***: fails for large non-gaussianity.
2. Normalizing flow-based nested[3]/Gaussianized bridge[4] sampling - ***Requires likelihood re-calculation***

1. John Skilling "Nested Sampling," 10.1063/1.1835238.
2. A. Heavens, et al 2017 arXiv:1704.03472 [stat.CO]
3. Nested sampling with normalizing flows for gravitational-wave inference, 10.1103/PhysRevD.103.103006
4. Jia, He; Seljak, Uroš, 2019 10.48550/arXiv.1912.06073

A normalizing flow

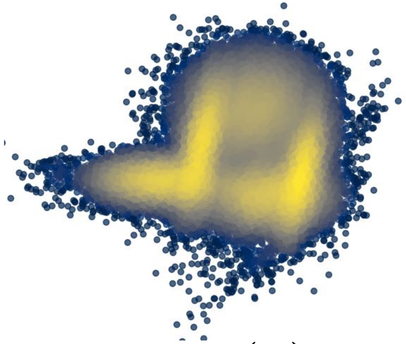
Flows solves for a bijective map b/ the *latent* Normal distribution and the *real* non-trivial distribution.

Known *latent* distribution



$$\boldsymbol{y} \sim \mathrm{n}(\boldsymbol{y})$$

Target *real* distribution

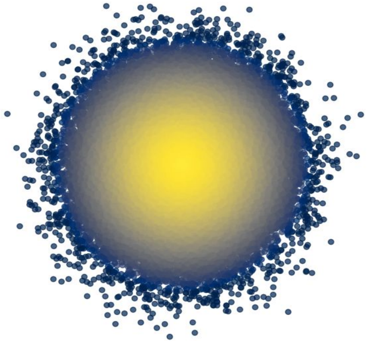


$$\boldsymbol{x} \sim \mathrm{p}(\boldsymbol{x})$$

A normalizing flow

Flows solves for a bijective map b/ the *latent* Normal distribution and the *real* non-trivial distribution.

Known *latent* distribution

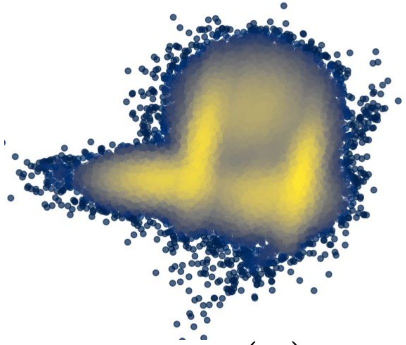


$\boldsymbol{y} \sim \mathrm{n}(\boldsymbol{y})$

$\boldsymbol{x} = \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{y})$

$\boldsymbol{y} = \mathbf{f}_{\boldsymbol{\phi}}^{-1}(\boldsymbol{x})$

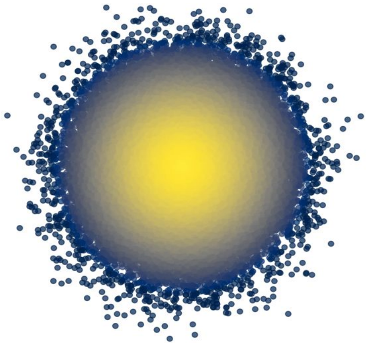
Target *real* distribution



$\boldsymbol{x} \sim \mathrm{p}(\boldsymbol{x})$

A normalizing flow

Flows solves for a bijective map b/ the *latent* Normal distribution and the *real* non-trivial distribution.

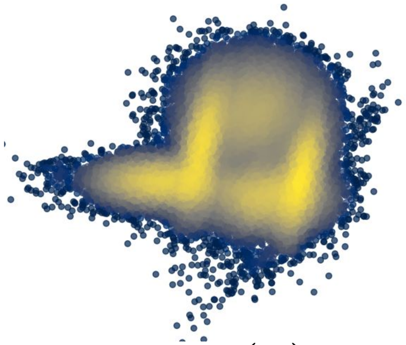


$$\boldsymbol{x} = \mathbf{f}_{\boldsymbol{\phi}}(\boldsymbol{y})$$

$$\Rightarrow$$

$$\Leftarrow$$

$$\boldsymbol{y} = \mathbf{f}_{\boldsymbol{\phi}}^{-1}(\boldsymbol{x})$$



$\boldsymbol{y} \sim \mathrm{n}(\boldsymbol{y})$ $\qquad$ $\boldsymbol{x} \sim \mathrm{p}(\boldsymbol{x})$

$$\text{Target distribution} \quad \mathrm{p}(\boldsymbol{x}) \mapsto \mathrm{q}_{\boldsymbol{\phi}}(\boldsymbol{x}) \quad \text{Flow prediction} \; = \mathrm{n}(\mathbf{f}_{\boldsymbol{\phi}}^{-1}(\boldsymbol{x})) \left| \det \frac{\partial \mathbf{f}_{\boldsymbol{\phi}}^{-1}}{\partial \boldsymbol{x}}(\boldsymbol{x}) \right|$$

Theory behind *floZ*

Evidence = normalization constant of likelihood x prior

Theory behind *floZ*

Evidence = normalization constant of likelihood x prior

Evidence Ground Truth

$$\boxed{Z \mapsto \zeta(\boldsymbol{x}, \boldsymbol{\phi}) = \frac{\hat{\mathrm{p}}(\boldsymbol{x})}{\mathrm{q}_{\boldsymbol{\phi}}(\boldsymbol{x})}}$$

Likelihood . prior

= *Unnormalized* posterior probability

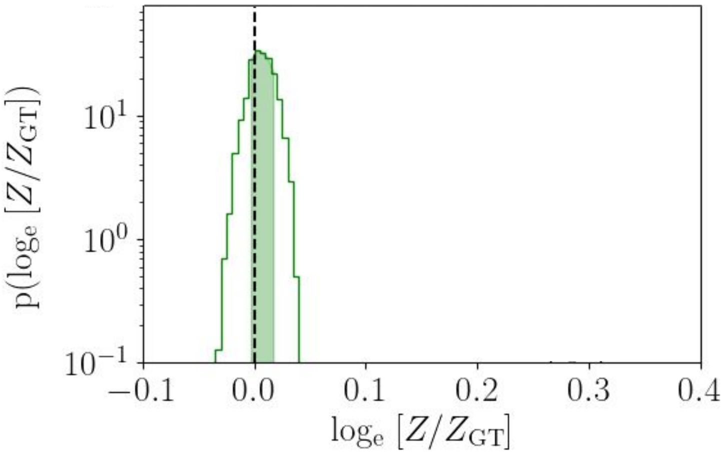
Evidence Estimation Distribution

floZ

= *Normalized* posterior probability

Expected output:

Evidence *distribution*



Ideally a delta function

Implementation:

Loss Terms

1. Normalizing flow loss:

floZ prediction

$$\mathcal{L}_1(\boldsymbol{\phi}) = -\mathrm{E}_{\mathrm{p}(\boldsymbol{x})}\left[\log(\mathrm{q}_{\boldsymbol{\phi}}(\boldsymbol{x}))\right]$$

Expectation over posterior samples

Implementation:
Loss Terms

1. Normalizing flow loss:

floZ prediction

$$\mathcal{L}_1(\phi) = -\mathrm{E}_{\mathrm{p}(\boldsymbol{x})}\left[\log(\mathrm{q}_{\boldsymbol{\phi}}(\boldsymbol{x}))\right]$$

Expectation over posterior samples

2. Reducing evidence estimation error:

$$\mathcal{L}_2(\phi) \simeq \log \sigma_{\mathfrak{h}}$$

Standard deviation of evidence estimation

Implementation:
Loss Terms

1. Normalizing flow loss:

floZ prediction

$$\mathcal{L}_1(\boldsymbol{\phi}) = -\mathrm{E}_{\mathrm{p}(\boldsymbol{x})}\left[\log(\mathrm{q}_{\boldsymbol{\phi}}(\boldsymbol{x}))\right]$$

Expectation over posterior samples

2. Reducing evidence estimation error:

$$\mathcal{L}_2(\boldsymbol{\phi}) \simeq \log \sigma_{\mathfrak{h}}$$

Standard deviation of evidence estimation

3. Identity evidence ratio of all pairs of samples:

Mean evidence ratio

$$\mathcal{L}_{3a}(\boldsymbol{\phi}) \ = \ |\log \mu_{\mathfrak{g}}|$$

Implementation:
Loss Terms

1. Normalizing flow loss:

floZ prediction

$$\mathcal{L}_1(\boldsymbol{\phi}) = -\mathrm{E}_{\mathrm{p}(\boldsymbol{x})}\left[\log(\mathrm{q}_{\boldsymbol{\phi}}(\boldsymbol{x}))\right]$$

Expectation over posterior samples

2. Reducing evidence estimation error:

$$\mathcal{L}_2(\boldsymbol{\phi}) \simeq \log \sigma_{\mathfrak{h}}$$

Standard deviation of evidence estimation

3. Identity evidence ratio of all pairs of samples:

Mean evidence ratio

$$\mathcal{L}_{3a}(\boldsymbol{\phi}) = |\log \mu_{\mathfrak{g}}|$$

4. Reducing evidence ratio error:

$$\mathcal{L}_{3b}(\boldsymbol{\phi}) = \log \sigma_{\mathfrak{g}}$$

Standard deviation of the ratio of evidence

Implementation:

Loss Terms

1. Normalizing flow loss:

floZ prediction

$\mathcal{L}_1(\phi) =$ **L₁** $q_\phi(x))]$

Expect *rior samples*

2. Reducing evidence estimation error:

$\mathcal{L}_2(\phi) \simeq 1$ **L₂**

viation of evidence estimation

3. Identity evidence ratio of all pairs of samples:

idence ratio

$\mathcal{L}_{3a}(\phi)$ **L₃ₐ**

4. Reducing evidence ratio error:

$\mathcal{L}_{3b}(\phi) =$ **L₃b**

Standard deviation of the ratio of evidence

Implementation:

Loss Scheduling

Solving the four losses simultaneously:

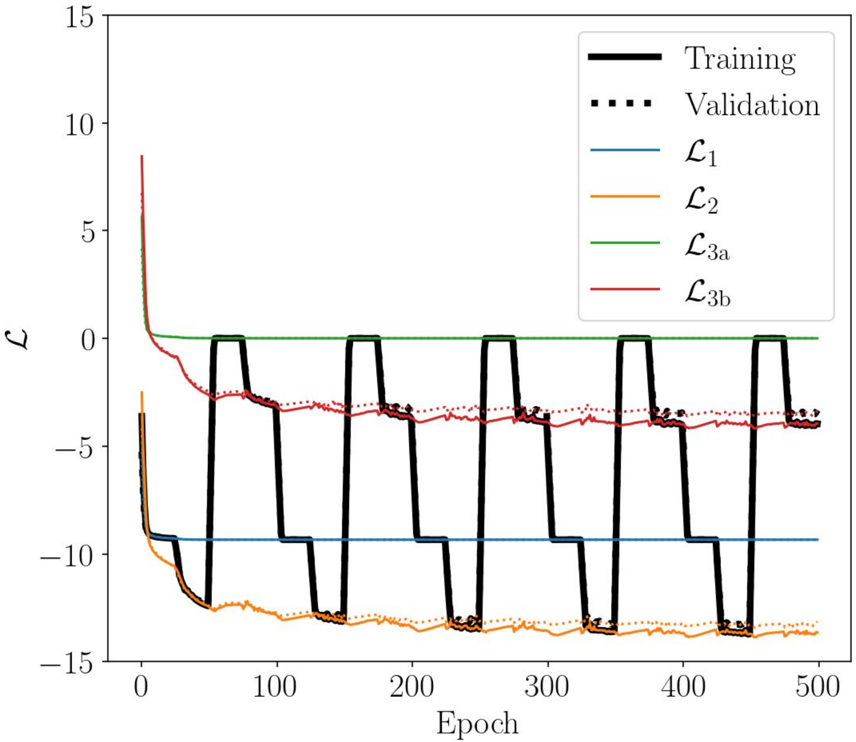
1) Weighted sum of losses.
2) Schedule the losses

Implementation:

Loss Scheduling

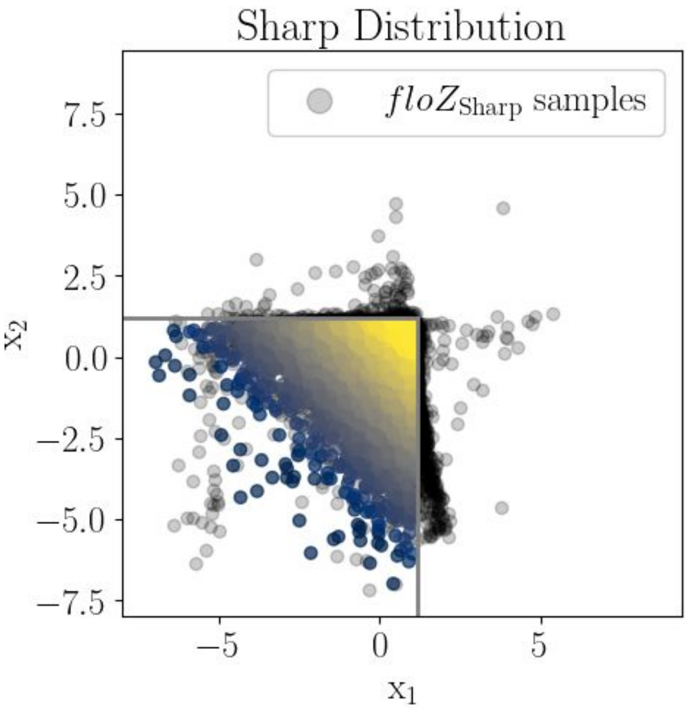
Solving the four losses simultaneously:

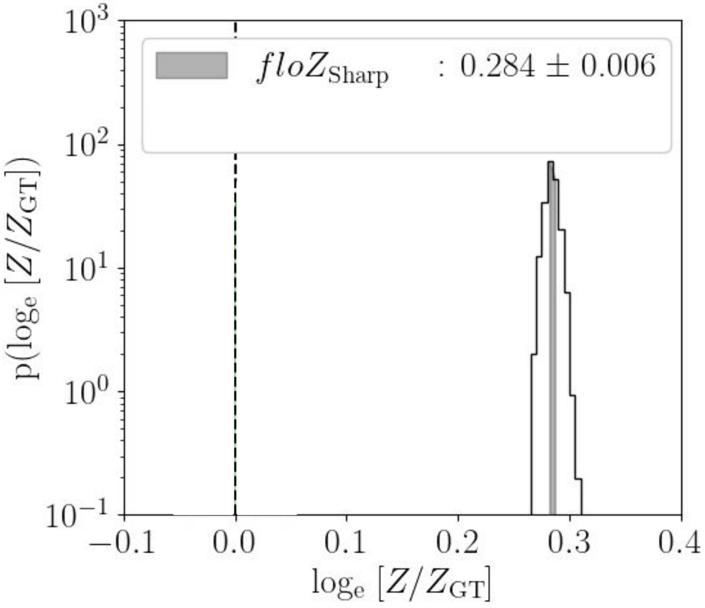
1) Weighted sum of losses.
2) ***Schedule the losses***



Implementation:

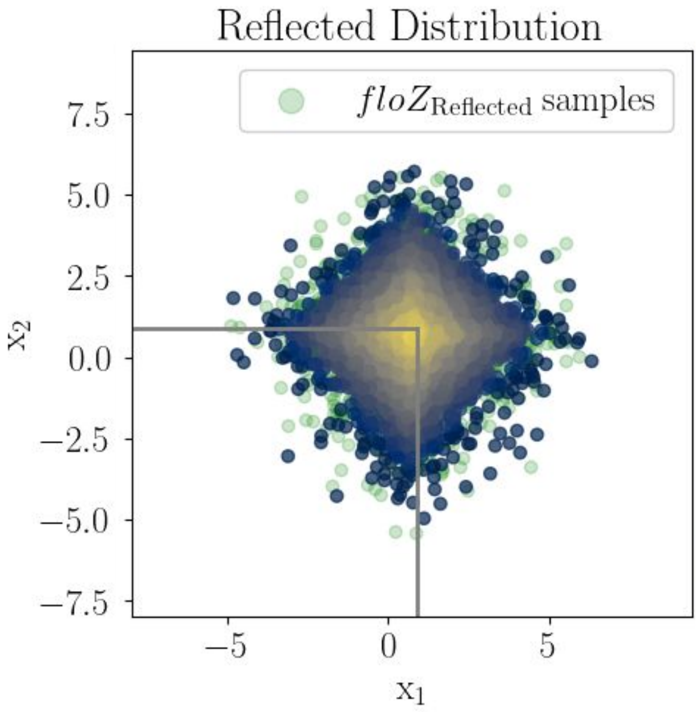
Dealing with sharp boundaries

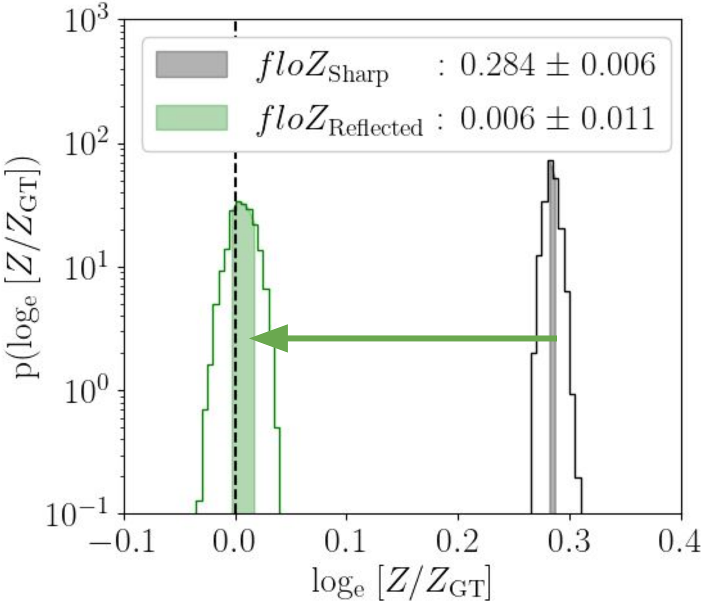




Implementation:

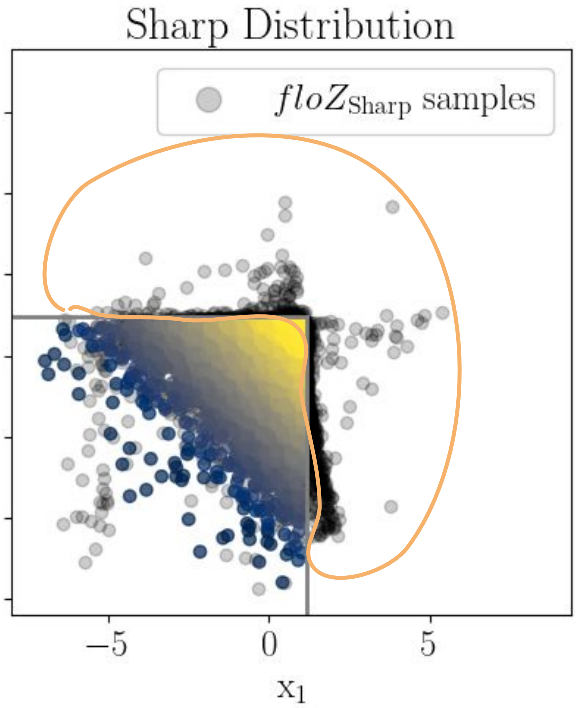
Dealing with sharp boundaries

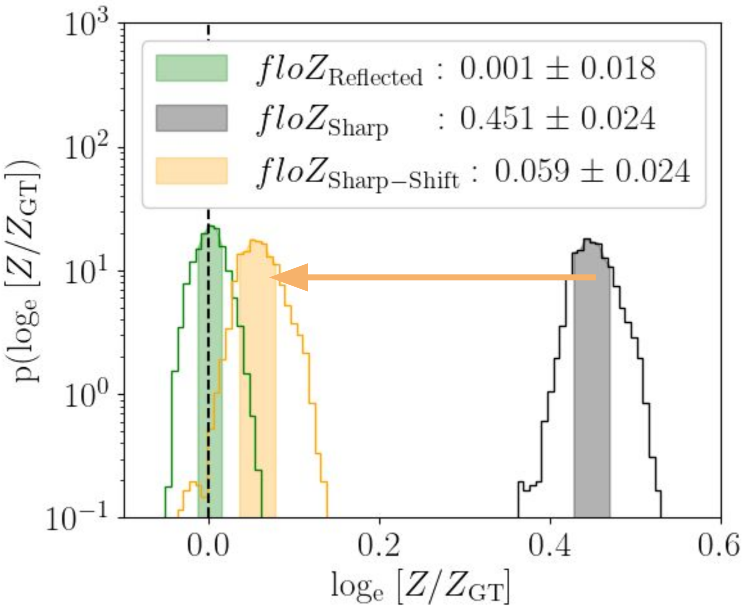




Alternatives?

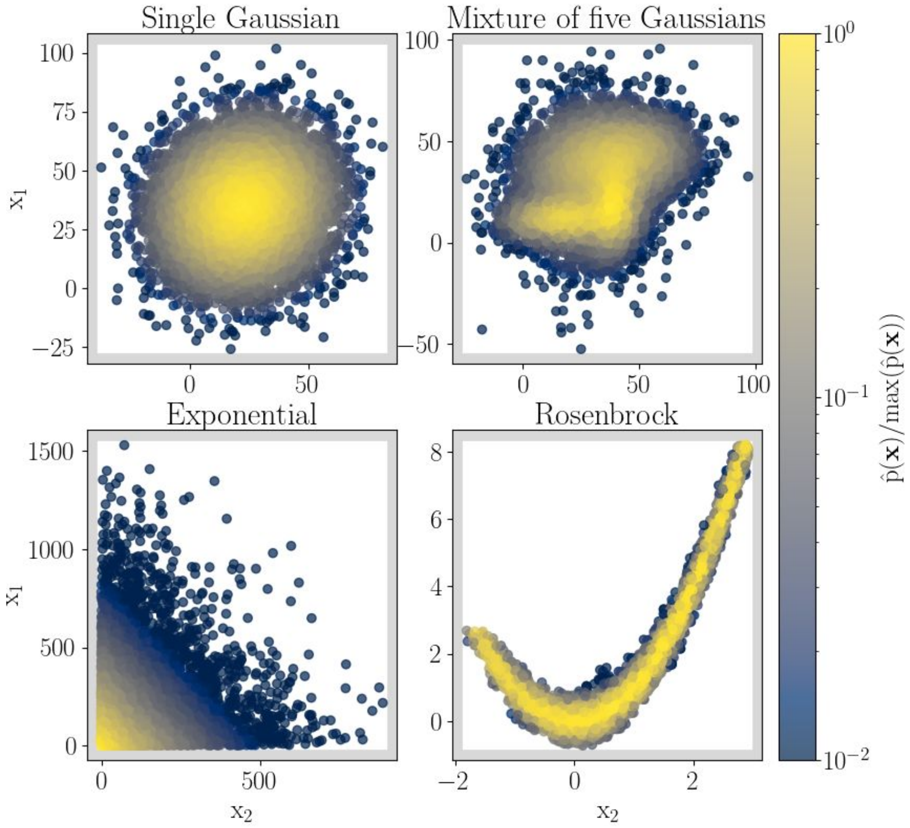
Reweighting by fraction of outliers





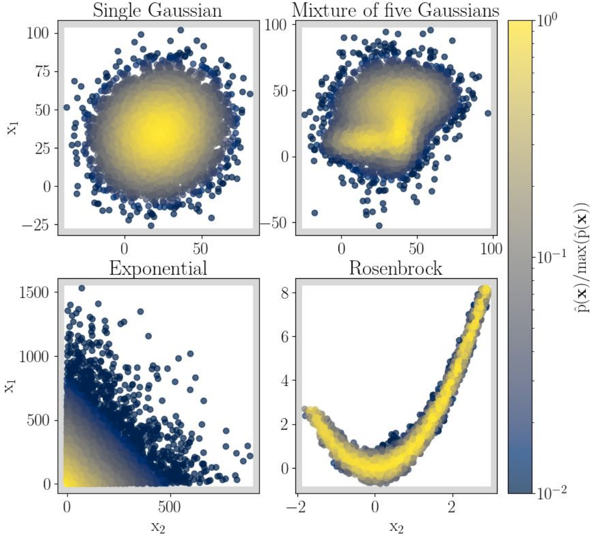
Distributions for benchmarking

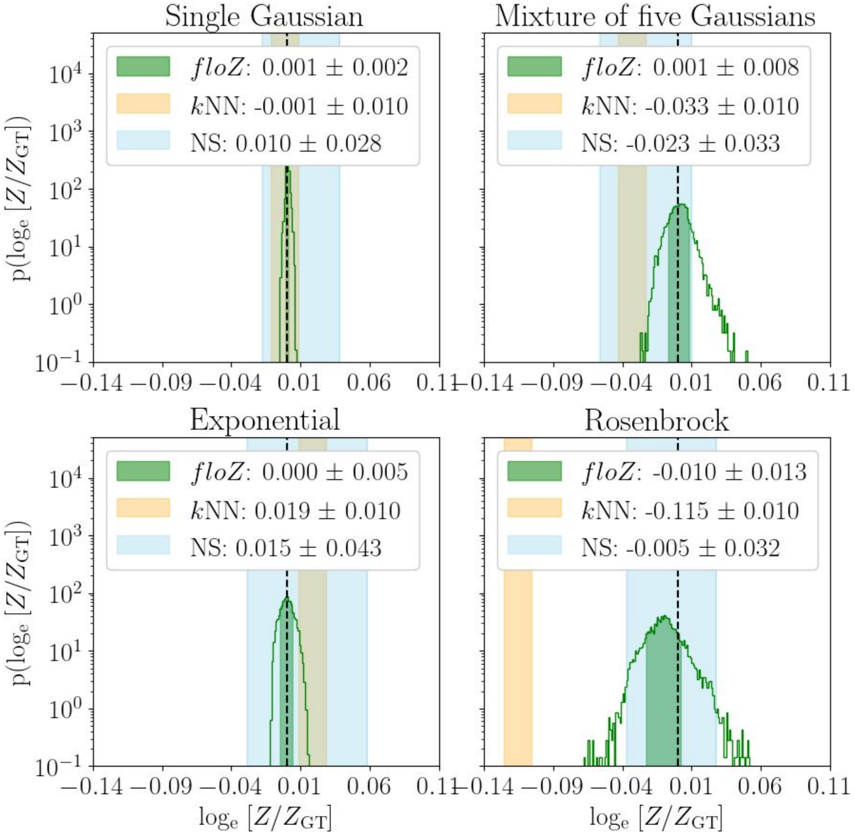
2, 10, 15 dimensions



Benchmarking w/ StateOfTheArt

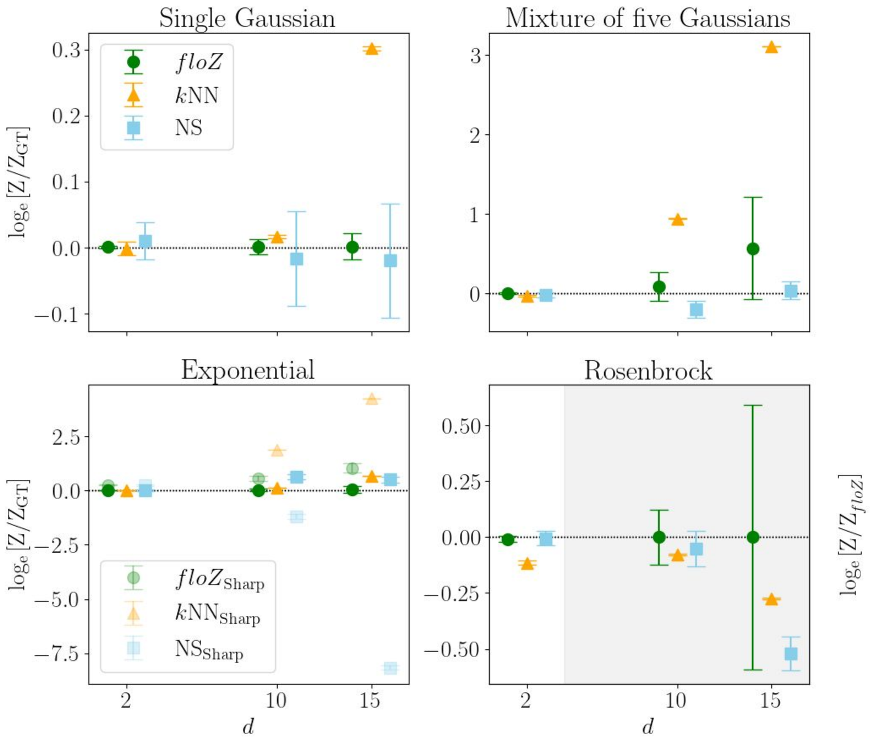
kNN: k-Nearest Neighbours
NS: Nested Sampling





Benchmarking w/ StateOfTheArt

4 Distributions x {2,10,15} Dimensions



- ***Accurate***:

  floZ and NS are in good agreement.

  Outperforms *k*NN

- ***Scalable***:

  15d require no more than 10^5 samples.

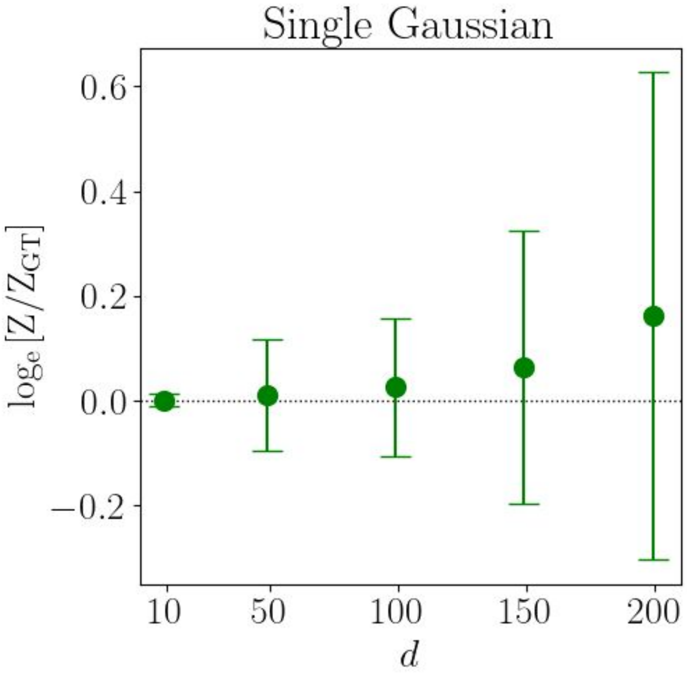
- **Rapid**

  15d results of ***floZ*** obtained in ~ 20min on an A100 GPU

High dimensional scalability

For the same number of samples (10^5) & model complexity.

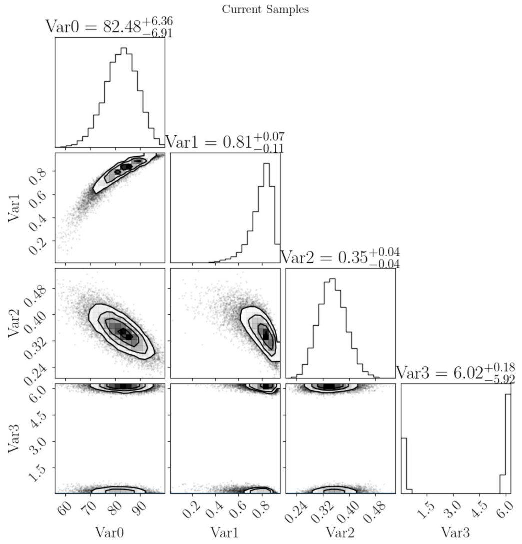
* For complex distributions, we need a combination of more samples, longer training time, and deeper networks.

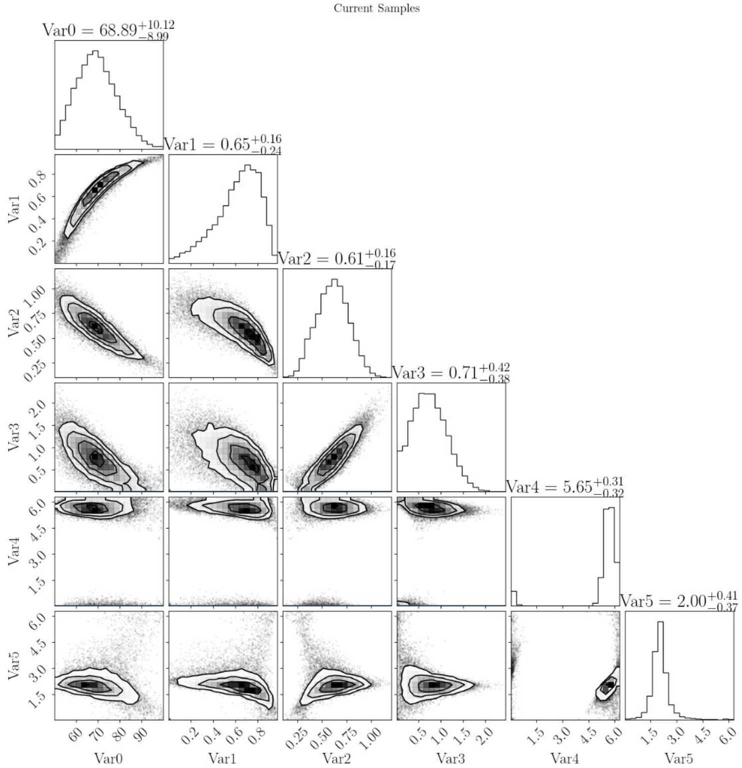


Applications: GW Ringdown

Bayes factor in favor of the presence of the higher 221 overtone in GW150914

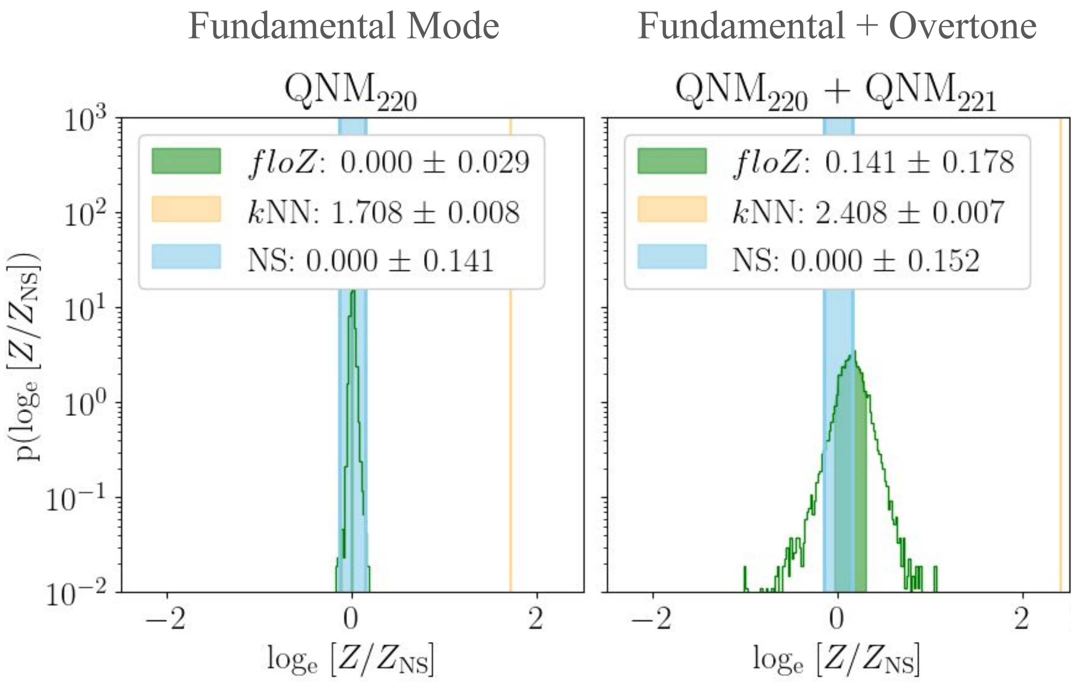
Fundamental Mode vs Fundamental Mode w/ Overtone





Applications: GW Ringdown

Bayes factor in favor of the presence of the higher 221 overtone in GW150914

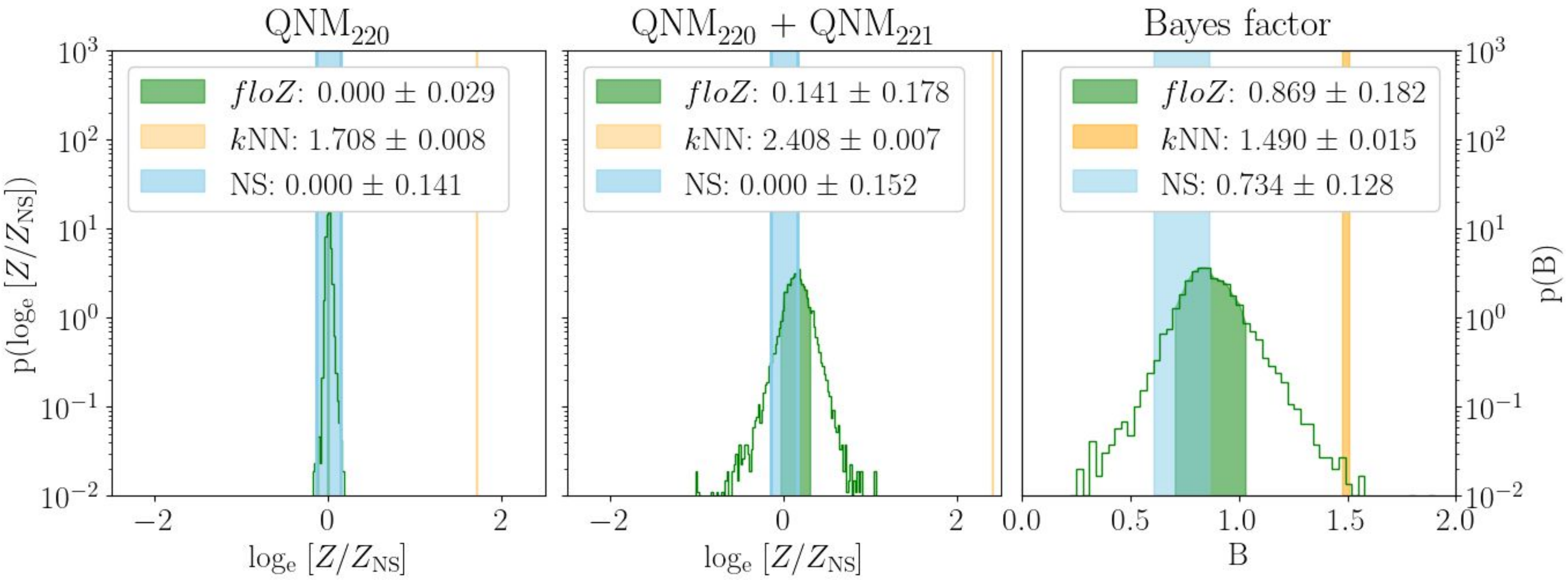


Samples from a nested sampler: CPNest [1]

1. W. Del Pozzo and J. Veitch, “CPNest: Parallel nested sampling.” Astrophysics source code library, record ascl:2205.021, May, 2022.

Applications: GW Ringdown

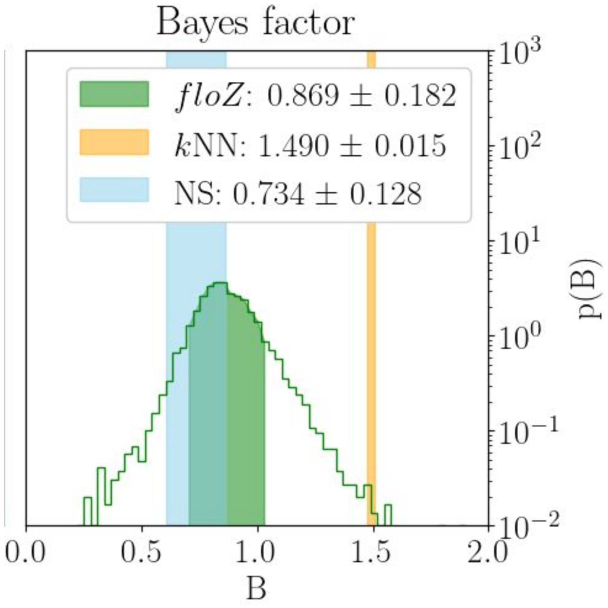
Bayes factor in favor of the presence of the higher 221 overtone in GW150914



Applications: GW Ringdown

Bayes factor in favor of the presence of the higher 221 overtone in GW150914

floZ estimates is compatible with nested sampling within their 1σ uncertainties.

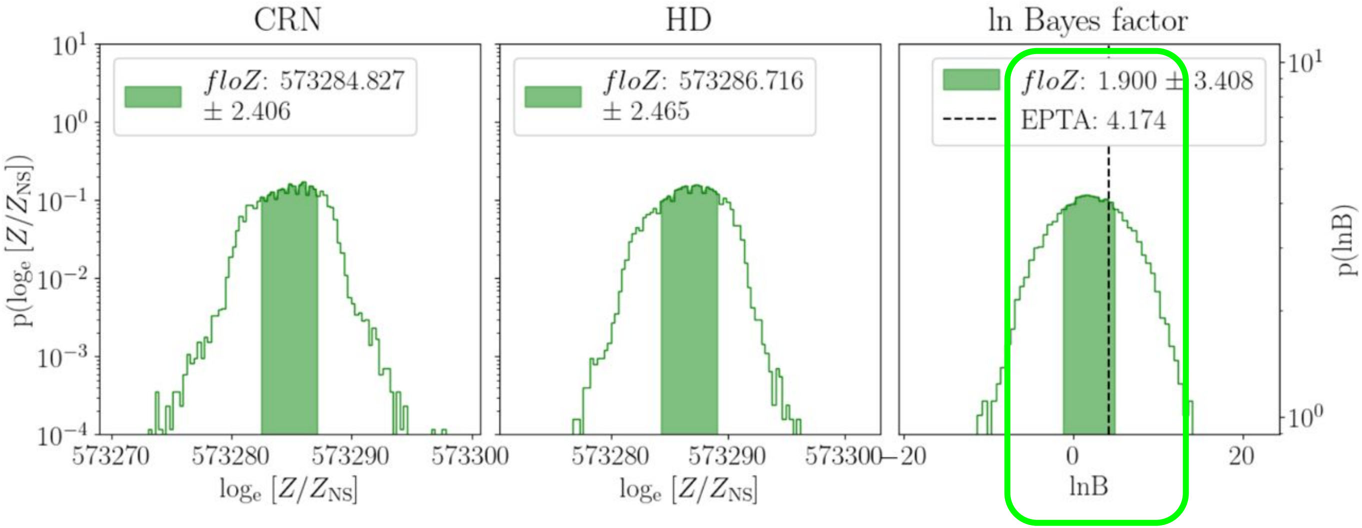


Applications: Pulsar Timing Array

Bayes factor in favor of the presence of Hellings-Downs relation in EPTA data

70 dimensional samples, with 1e5 samples.

floZ estimates is compatible with EPTA results within the 1σ uncertainties.
Very non-gaussian distribution → Need more samples (ongoing analysis)



Samples provided by the EPTA collaboration

Convergence Test

How do we know that the flow is correct?

Convergence Test

How do we know that the flow is correct?

μ - σ plot

Mapping to Latent space should be a gaussian